AFOSR Final Technical Report

Covering the period 1 Jul 95 to 31 Dec 98

Project: **Bayesian Network Models for Pattern and Plan Recognition**

Grant Number: **F49620-95-1-0365 (AASERT supplement)**

Principal Investigator: **Michael P. Wellman**

University of Michigan
Artificial Intelligence Laboratory
1101 Beal Avenue
Ann Arbor, MI 48109-2110

Abstract

The aim of this AASERT supplement was to investigate the use of Bayesian networks to capture structural regularities in domains for pattern recognition. We focused on *plans* as a particular form of pattern, where the recognition elements are actions and their effects. Plans are distinguished by the fact that they are generated by agents to serve some objectives, and this causal relationship can be exploited in developing specific models to support the plan recognition task.

This AASERT supplement augmented our AFOSR project that covered more generally the scope of dynamic decision making under uncertainty. It primarily supported the graduate studies of David Pynadath, who successfully completed his dissertation (Pynadath 1999) in February 1999.

The specific results of this project comprised several advances in plan recognition under uncertainty, most notably: (1) a general Bayesian framework for plan recognition, (2) a generalization of techniques for probabilistic context-free grammars based on encoding the space of parse trees in Bayesian networks, (3) a new representation, probabilistic state-dependent grammars, exploiting the advantages of state-based and grammatical approaches, and (4) demonstrations of the new techniques in simplified versions of the application domains of highway traffic and air combat.

REPORT DOCUMENTATION PAGE

AFRL-SR-BL-TR-00-
0254

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
16 Jun 00

3. ...
Final Technical Report 1 Jul 95 to 31 Dec 98

4. TITLE AND SUBTITLE
Bayesian Network Models for Pattern and Plan Recognition

5. FUNDING NUMBERS
F49620-95-1-0365

6. AUTHOR(S)
John Rushby

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Michigan
Artificial Intelligence Laboratory
1101 Beal Avenue
Ann Arbor MI 48109-2110

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NM
801 N. Randolph St, Rm 732
Arlington, VA 22203-1977

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
F49620-95-1-0365

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The aim of this AASERT supplement was to investigate the use of Bayesian networks to capture structural regularities in domains for pattern recognition. We focused on plans as a particular form of pattern, where the recognition elements are actions and their effects. Plans are distinguished by the fact that they are generated by agents to serve some objectives, and this causal relationship can be exploited in developing specific models to support the plan recognition task. This AASERT supplement augmented our AFOSR project that covered more generally the scope of dynamic decision making under uncertainty. It primarily supported the graduate studies of David Pynadath, who successfully completed his dissertation (Pynadath 1999) in February 1999. The specific results of this project comprised several advances in plan recognition under uncertainty, most notably: (1) a general Bayesian framework for plan recognition, (2) a generalization of techniques for probabilistic context-free grammars based on encoding the space of parse trees in Bayesian networks, (3) a new representation, probabilistic state-dependent grammars, exploiting the advantages of state-based and grammatical approaches, and (4) demonstrations of the new techniques m simplified versions of the application domains of highway traffic and air combat.

14. SUBJECT TERMS

15. NUMBER OF PAGES
10

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

1. Overview

The aim of this project was to identify general principles and develop concrete techniques for plan recognition under uncertainty. We exploited two threads of prior work bearing on this problem. First, we extended and generalized grammatical approaches, such as those based on probabilistic context-free grammars. Grammatical representations capture generative structure in sequential patterns (such as plans), are well understood theoretically, and have a productive history of use in recognition, particularly of language. Second, we employed Bayesian networks for general representation of uncertain relationships, and particularly for capturing dependence structure in state representation. The synthesis of these two threads form the major theme for the line of work pursued in this project.

2. Pattern and Plan Recognition

The problem of *plan recognition* is to induce the plan of action driving an agent's behavior, based on partial observation of its behavior up to the current time. Deriving the underlying plan can be useful for many purposes—predicting the agent's future behavior, interpreting its past behavior, or generating actions designed to influence the plan itself. Researchers in AI have studied plan recognition for several kinds of tasks, including discourse analysis (Grosz and Sidner 1990), collaborative planning (Huber and Durfee 1993), and adversarial planning (Azarewicz et al. 1989). These works have employed a great variety of reasoning techniques, operating on similarly various plan representations and adopting varied assumptions about observability.

The common theme underlying these diverse motivations and approaches is that the object to be induced is a *plan*, and that this plan is the cause of observed behavior. If there is anything special about the task of plan recognition as opposed to recognition in general, it must be due to special properties of plans: how they are constituted, and how they cause the behavior we observe and wish to predict, interpret, and influence.

We can distinguish plan recognition from uncertain reasoning in general by noting two special features of plans. First, plans are *structured linguistic objects*. Plan languages considered in AI research range from simple sequences of action tokens to general-purpose programming languages. In either case, the recognizer can and should exploit the structure of plans in inducing them from partial observations of the actions comprising the plan. Another way to say this is that plans are descriptions of action *patterns*, and therefore any general pattern-recognition technique is automatically a plan recognition technique for the class of plans corresponding to the class of patterns associated with the given technique.

The second special feature of plans is that they are *rational constructions*. They are synthesized by a rational agent with some beliefs, preferences, and capabilities, that is, a *mental state*. Knowing the agent's mental state and its rationality

properties strongly constrains the possible plans it will construct. (The degree of constraint depends on the power of the rationality theory we adopt.) The rational origin of plans is what distinguishes plan recognition from pattern recognition. If the observations available include evidence bearing on the beliefs, preferences, and capabilities of the agent, then the recognizer should combine this with evidence from the observed actions in reasoning about the entire plan.

Our first step in this project was to elucidate (Pynadath and Wellman 1995) a general Bayesian framework for plan recognition. Our basic approach is similar to that of Charniak and Goldman (Charniak and Goldman 1993), elaborating and departing in some respects, less well-developed in others. We describe the high-level idea below; for a more complete description and some specific developments of the technique see the cited papers.

Our framework is *Bayesian* in that we start from a causal theory of how the agent's mental state causes its plan and executing its plan causes activity, and reason from observed effects to underlying causes. Our recognizer has uncertain *a priori* knowledge about the agent's mental state, the world state, and the world's dynamics, which can be summarized (at least in principle) by a probability distribution. It then makes partial observations about the world, and uses this evidence to induce properties of the agent and its plan.

We begin with a model of the planning agent operating in the world. As it begins planning, the agent has a certain mental state, consisting of its preferences (e.g., goals), beliefs (e.g., about the state of its environment), and capabilities (e.g., available actions). We assume the actual planning process to be some rational procedure for generating the plan that will best satisfy the agent's preferences based on its beliefs, subject to its capabilities. This plan then determines (perhaps with some uncertainty) the actions taken by the agent in the world.

Once we have accounted for the agent's plan-generation process, we need to consider the effects of the plan's execution. In many plan-recognition domains, the external observer finds the agent's actions inaccessible. In such cases, the recognizer observes actions only indirectly, via their effects on the world (which themselves are typically only partially observable). These restricted observations then form the basis of inference.

Thus, observations of the state of the world provide two types of evidence about the plan. First, the world influences the agent's initial mental state, which provides the *context* for plan generation. Second, changes in the world state reflect the effects of the agent's actions, which *result* from executing its plan.

3. Bayesian Networks for Plan Recognition

To perform plan recognition tasks, we generate a Bayesian network representing the causal planning model and use it to support evidential reasoning from observations to plan hypotheses. The structure of the Bayesian network is based on the framework depicted in Figure 1. That diagram can itself be viewed as a Bayesian network, albeit with rather broad random variables. To make this operational, we replace each component of the model with a

subnetwork that captures intermediate structure for the particular problem. The limited connections among the subnetworks reflect the dependency structure of our generic planning model.

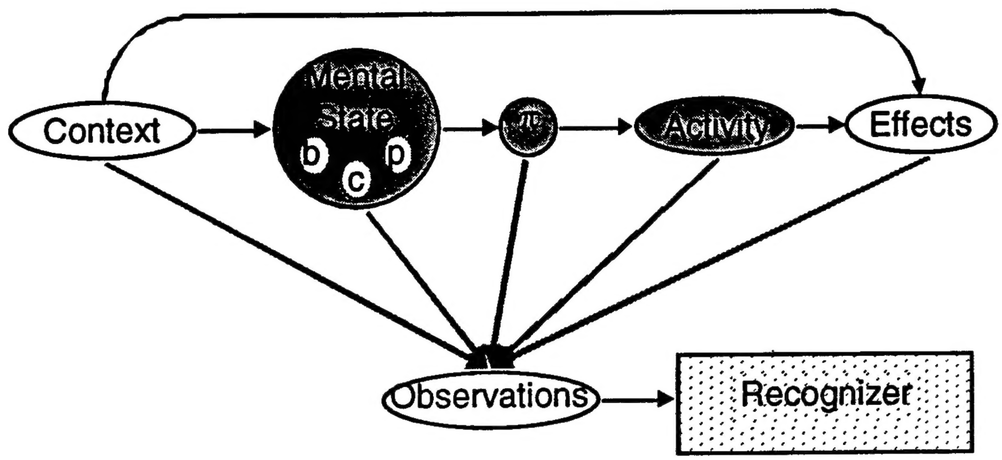

Figure 1: Bayesian plan recognition framework.

The framework as described above is of course very general. We have explored particular instances of the approach, specifically looking at the issue of modeling context in the domain of traffic monitoring (Pynadath and Wellman 1995). In subsequent work, we investigated more deeply the problem of modeling the planning process. In doing so, we need to adopt particular assumptions about the plans generated, and determine an effective recognition strategy.

3.1 Probabilistic Context-Free Grammars

Our approach has been to treat plan generation as a structured stochastic process, and recognition as the task of answering queries about events in the generation of particular observations. Our first deep study adopted the generative model of *probabilistic context-free grammars* (PCFGs), a well-studied and commonly applied model for pattern recognition (Charniak 1993; Wetherell 1980). Interpreting a string of observations generated from a grammar is known as *parsing*, and the general recognition problem can be cast in terms of queries about the parse. For PCFGs, efficient algorithms have been developed for several useful types of queries (i.e., calculating the probability of a given string, or finding the most likely parse). However, for other queries potentially useful in plan recognition, only brute-force enumeration is available.

To extend this approach, we have shown (Pynadath and Wellman 1996; Pynadath and Wellman 1998) how to construct a Bayesian network to represent the distribution of parse trees induced by a given PCFG. The network structure mirrors that of the chart in a standard parser, and is generated using a similar dynamic-programming approach. By augmentations of the network, we can

relax the context-free restriction of the grammar in a controlled way, admitting important context-sensitivities without invalidating the inferences drawn by the recognizer.

This method generalizes the class of queries that can be answered in several ways:

(1) allowing missing tokens in a sentence or sentence fragment,

(2) supporting queries about intermediate structure, such as the presence of particular nonterminals, and

(3) flexible conditioning on a variety of types of evidence.

We direct the reader to our published work for discussion of the technical details of our algorithm. In these documents, we present an algorithm for constructing Bayesian networks from PCFGs, and show how queries or patterns of queries on the network correspond to interesting queries on PCFGs.

3.2 Air Force Plan Recognition Application

The generalized pattern-recognition procedure is potentially applicable to a wide range of Air Force problems involving interpreting uncertain or incomplete observations. One example comprises problems of plan recognition, where the aim is to interpret or predict the actions of an observed agent (friend or foe), based on uncertain observations of its action thus far.

One of the more common representations for planning structures used in plan-recognition research is an action decomposition hierarchy, sometimes called an *event tree* (Kautz and Allen 1986). Event trees and other variants of hierarchies map easily to context-free grammars, and indeed the parsing approach to recognition has previously been proposed (for the deterministic case) by Vilain (1990). By extending the event-tree model to include probabilities, we provide a basis for distinguishing among equally possible but unequally plausible explanations of the observations. As Charniak and Goldman (1993) (among others) have argued, this is a critical requirement for any useful plan recognition algorithm.

In air-combat scenarios, for example, we can model the behavior of a fighter plane to allow tracking and prediction of its actions. The probabilistic event tree could include information about possible specializations of its general mission (e.g. fly to target, intercept enemy plane), as well as decompositions of plans into subplans (e.g. employ weapons, evade, chase) or observable actions (e.g. start turning, stop turning, maintain current heading). An example event tree for an air-to-air combat scenario (borrowed from Tambe and Rosenbloom (1995)) is presented in Figure 2.

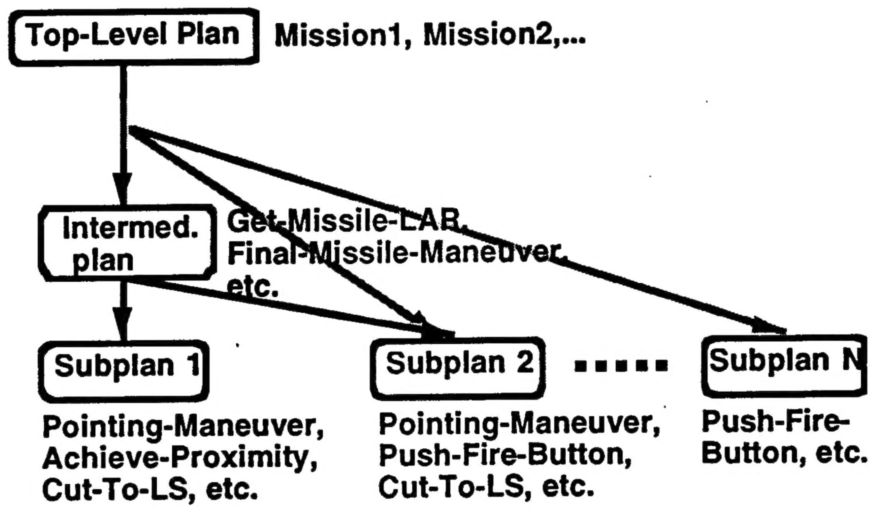

Figure 2: An example event tree from air combat domain.

We can then translate this event tree representation into a probabilistic grammar whose rules correspond to the plan specializations and decompositions. The algorithm mentioned above can use this grammar to generate a Bayesian network corresponding to the probability distribution over the possible behavior of the tracked plane. This network would support a wide variety of useful queries, using the traditional methods of evidence propagation to compute the relevant probabilities. In the air-combat example, a pilot may wish to determine whether a nearby enemy plane is about to launch a missile, or is merely flying to another target. The Bayesian network can provide the probability of either subplan, conditioned on whatever behavior has been observed so far. These probabilities, along with the different implications of the two cases, can aid the pilot in choosing the correct course of action.

4. Probabilistic State-Dependent Grammars

One drawback of using PCFGs for plan recognition (as well as most other extant approaches), is that they require maintaining the entire history of observations as context for subsequent plan recognition queries. Whereas this is unavoidable in general, it may well be possible to employ graphical modeling techniques to exploit whatever independence exists to support practical inference.

Our investigations culminated in the development of *probabilistic state-dependent grammars* (PSDGs), a hybrid representation based on PCFGs augmented with dependencies of production probabilities on underlying states (Pynadath and Wellman 2000). The state evolution is defined by transition probabilities, represented in dynamic Bayesian networks (DBNs) (Kjærulff 1992). Although any PSDG can be represented as a PCFG, the PSDG representation may be exponentially more compact when the state space is highly structured.

4.1 Example: Highway Traffic

Consider the following PSDG, representing a simplified model of driving plans.

0)	Drive	$\rightarrow$	Stay Drive	$(p_0(q) = \ldots)$
1)	Drive	$\rightarrow$	Left Drive	$(p_1(q) = \{0$ if Lane(q) = left-lane $\ldots)$
2)	Drive	$\rightarrow$	Right Drive	$(p_2(q))$
3)	Drive	$\rightarrow$	Pass Drive	$(p_3(q))$
4)	Drive	$\rightarrow$	Exit	$(p_4(q))$
5)	Drive	$\rightarrow$	Left Right	$(p_5(q))$
6)	Drive	$\rightarrow$	Right Left	$(p_6(q))$

The state includes the observable features of the driver's position and speed, as well as the position and speeds of other cars on the highway. The state also includes aspects of the driver's mental state, such as the agent's preferences about driving speed, distance from other cars, intended exit, etc. We can explore the generation of the parse tree of Figure 3 (corresponding to one possible instance of the agent's plan generation and execution) to illustrate the interactions between the plan and state models. The pictures across the bottom of the diagram represents the observable portion of state at that point of the parse tree. The darker rectangle (blue if reading in color) is the driver whose planning process we are trying to recognize. The lighter rectangles (green) are the other cars on the highway that the driver of interest must consider when planning.

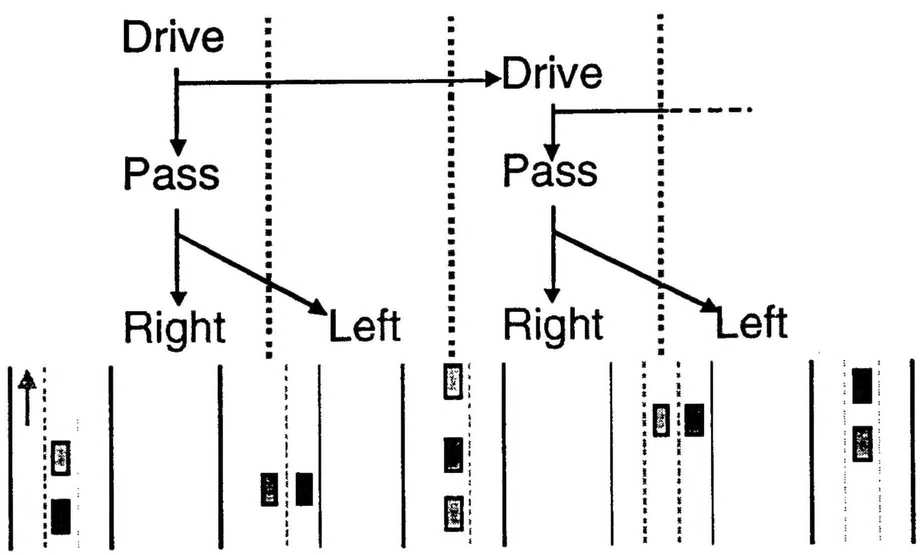

Figure 3: A parse tree for the simple highway example.

In this scenario, the driver passes two cars, both times on the right. The PSDG formalism makes it easy to specify how the driver's decision to pass, and on which side, may depend on situational factors. This is accomplished by

conditioning the production probabilities in the grammar on the underlying state. For example, the decision to pass might be based on desired speed and speed of the car ahead, whereas choice of side might depend on the presence of cars in other lanes.

Our implemented traffic PSDG has 14 nonterminal symbols (plans), 7 terminal symbols (actions), and 15 state features (with the mean state space size being 431 elements). Three of these state features correspond to aspects of the driver's mental state (preferred speed, intended exit, aggressiveness); the rest of the state features are completely observable. There are a total of 40 productions with a mean length of two symbols. We also implemented a PSDG representation for an air combat domain based on an existing specification (Tambe and Rosenbloom 1996) using SOAR productions (Laird et al. 1987).

4.2 PSDG Inference

Although we can perform inference on a given PSDG with a finite state space by generating the corresponding PCFG and using PCFG inference algorithms, the explosion in the size of the symbol space can lead to prohibitive costs. In addition, existing PCFG algorithms cannot handle most plan-recognition queries.

We can potentially perform inference by generating a DBN representation of a PSDG distribution. The definition of the PSDG language model supports an automatic DBN generation algorithm. The resulting DBN supports queries over the symbol, production, and state random variables. Unfortunately, the complexity of DBN inference is likely to be impractical for most PSDGs, where the belief state must represent the entire joint distribution over all possible combinations of state and parse tree branches. For instance, for the complete PSDG representation of the traffic domain, the DBN belief state would have more than 10^{25} entries.

Instead, we have designed and implemented inference algorithms that exploit the particular structure of the PSDG model to answer a set of queries more restricted than that provided by DBNs. These algorithms use a compact belief state to answer queries based on observations of the state variables. At time t, the recognizer observes some or perhaps all of the features of the state, Q^t. Based on this evidence, the algorithm computes posterior probabilities over the individual state elements, as well as posterior probabilities over the possible plans and productions that the agent executed at time $t-1$. The algorithm then computes the posterior probabilities over the plans and productions that the agent will select at time t, as well as updating the recognizer's belief state.[1]

---

[1] A pseudocode description of the algorithm is available online at http://www.isi.edu/~pynadath/Research/PSDG. Both the pseudocode and proofs of correctness are presented in the dissertation (Pynadath 1999).

5. Conclusion

In this report, we have documented a series of advances in techniques for plan recognition under uncertainty. Starting from a general Bayesian framework, we explored the use of graphical models as well as existing grammatical formalisms. We combined the advantages of both approaches in a method for generating Bayesian networks from PCFGs. We then improved on that approach by defining a grammatical formalism—probabilistic state-dependent grammars—conducive for modeling plan recognition domains. We then developed inference algorithms designed to support plan recognition queries on PSDGs, and demonstrated their applicability in highway traffic and air combat domains.

References

Azarewicz, J., G. Fala, and C. Heithecker. 1989. Template-based multi-agent plan recognition for tactical situation assessment. In *Sixth Conference on Artificial Intelligence Applications*.

Charniak, Eugene: 1993. *Statistical Language Learning*. Cambridge, MA: MIT Press.

Charniak, Eugene, and Robert P. Goldman. 1993. A Bayesian model of plan recognition. *Artificial Intelligence* 64:53-79.

Grosz, B. J., and C. L. Sidner. 1990. Plans for discourse. In *Intentions in Communication*, edited by P. R. Cohen, J. Morgan and M. E. Pollack: MIT Press.

Huber, Marcus H., and Edmund H. Durfee. 1993. Observational uncertainty in plan recognition among interacting robots. In *IJCAI Workshop on Dynamically Interacting Robots*, Chambery, France.

Kautz, Henry A., and James F. Allen. 1986. Generalized plan recognition. In *National Conference on Artificial Intelligence*, Philadelphia.

Kjærulff, Uffe. 1992. A computational scheme for reasoning in dynamic probabilistic networks. In *Eighth Conference on Uncertainty in Artificial Intelligence*, Stanford, CA.

Laird, John E., Allen Newell, and Paul S. Rosenbloom. 1987. SOAR: An architecture for general intelligence. *Artificial Intelligence* 33:1-64.

Pynadath, David V. 1999. Probabilistic Grammars for Plan Recognition. PhD dissertation, Electrical Engineering and Computer Science, University of Michigan.

Pynadath, David V., and Michael P. Wellman. 1995. Accounting for context in plan recognition, with application to traffic monitoring. In *Eleventh Conference on Uncertainty in Artificial Intelligence*, Montréal.

Pynadath, David V., and Michael P. Wellman. 1996. Generalized queries on probabilistic context-free grammars. In *National Conference on Artificial Intelligence*, Portland, OR.

Pynadath, David V., and Michael P. Wellman. 1998. Generalized queries on probabilistic context-free grammars. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20:65-77.

Pynadath, David V., and Michael P. Wellman. 2000. Probabilistic state-dependent grammars for plan recognition. In *Sixteenth Conference on Uncertainty in Artificial Intelligence*, Stanford, CA.

Tambe, Milind, and Paul S. Rosenbloom. 1995. RESC: An approach for real-time, dynamic agent tracking. In *Fourteenth International Joint Conference on Artificial Intelligence*, Montreal.

Tambe, Milind, and Paul S. Rosenbloom. 1996. Event tracking in a dynamic multi-agent environment. *Computational Intelligence* 12:499-521.

Vilain, Marc. 1990. Getting serious about parsing plans. In *National Conference on Artificial Intelligence*, Boston.

Wetherell, C. S. 1980. Probabilistic languages: A review and some open questions. *Computing Surveys* 12 (4):361-379.